



## Development of a prototype smart home intelligent lighting control architecture using sensors onboard a mobile computing system

Tang, S., Kalavally, V., Ng, K., & Parkkinen, J. (2017). Development of a prototype smart home intelligent lighting control architecture using sensors onboard a mobile computing system. *Energy and Buildings*, 138, 368-376. <https://doi.org/10.1016/j.enbuild.2016.12.069>

[Link to publication record in Ulster University Research Portal](#)

**Published in:**  
Energy and Buildings

**Publication Status:**  
Published (in print/issue): 01/03/2017

**DOI:**  
[10.1016/j.enbuild.2016.12.069](https://doi.org/10.1016/j.enbuild.2016.12.069)

**Document Version**  
Author Accepted version

### General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

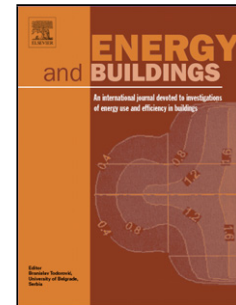
### Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [pure-support@ulster.ac.uk](mailto:pure-support@ulster.ac.uk).

## Accepted Manuscript

Title: Development Of A Prototype Smart Home Intelligent Lighting Control Architecture Using Sensors Onboard A Mobile Computing System

Author: Samuel Tang Vineetha Kalavally Ng Kok Yew Jussi Parkkinen



PII: S0378-7788(16)31997-1  
DOI: <http://dx.doi.org/doi:10.1016/j.enbuild.2016.12.069>  
Reference: ENB 7248

To appear in: *ENB*

Received date: 11-8-2016  
Revised date: 21-12-2016  
Accepted date: 22-12-2016

Please cite this article as: Samuel Tang, Vineetha Kalavally, Ng Kok Yew, Jussi Parkkinen, Development Of A Prototype Smart Home Intelligent Lighting Control Architecture Using Sensors Onboard A Mobile Computing System, *Energy & Buildings* (2016), <http://dx.doi.org/10.1016/j.enbuild.2016.12.069>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Development Of A Prototype Smart Home Intelligent Lighting Control Architecture Using Sensors Onboard A Mobile Computing System

Samuel Tang<sup>a</sup>, Vineetha Kalavally<sup>a</sup>, Ng Kok Yew<sup>a,\*</sup>, Jussi Parkkinen<sup>b</sup>

<sup>a</sup>Department of Electrical and Computer Systems Engineering, Monash University Malaysia

<sup>b</sup>School of Computing, University of Eastern Finland

## Abstract

As smartphones become increasingly powerful and ubiquitous, integrating them into intelligent lighting systems can boost both convenience and energy efficiency. This paper presents an intelligent lighting system prototype with enhanced security features for smart homes. The custom-built Android mobile application made use of the onboard ambient light sensor to run a novel closed-loop feedback algorithm to implement daylight harvesting. A cost analysis shows that the whole system setup is slightly cheaper than commercial products and due to its daylight harvesting capabilities, has potential for monetary savings in the long run, outperforming current commercial products.

**Keywords:** Smart home, intelligent lighting, daylight harvesting, LED lighting system, mobile device, smartphone

## 1. Introduction

The Internet of Things (IoT) is an emerging concept of everyday objects that are interconnected with each other [1]. With this comes the concept of smart homes where consumer electronic products and systems are automated and can be controlled easily by the users improving convenience, comfort, efficiency and security [1]. Within the many subsystems of a smart home, lighting plays a profoundly big role in our daily lives, not just at night, but even during the day, where artificial lighting is used to light up the indoors. The uptake of light emitting diodes (LED) as the main light source in the residential segment is forecasted to be at almost 50% in 2016 and over 70% in 2020 by McKinsey [2], showing a high adoption rate of solid state LED lighting by consumers due to its high energy efficiency and long lifespan. With the advancement of LED technologies, much more technologically complicated and challenging controls can be performed on them compared to halogen bulbs or fluorescent lamps. As a result, improvements in the standards of living in terms of convenience, ambience, customizability and power savings by using artificial lighting can be achieved through an LED-based intelligent lighting system. For example, the brightness and color of the LED luminaires can be controlled, functionalities such as dimming can be enabled to save power or lighting color can be changed to suit the occasion, mood or situation. In a report released by the International Energy Agency in 2015, artificial lighting accounted for 15% of energy consumption in residential buildings [3]. Daylight harvesting which is a method in which daylight is used to offset the

amount of electric energy needed to light up a space can save up to 27% [4] or even 40% [5] of lighting power in areas that receive a significant amount of daylight.

Seeing great potential in smart lighting, many industrial companies have taken up the challenge to create commercial products, such as Philips Hue, OSRAM Lightify, and LIFX. Though these are the leading smart lighting systems in the market, they are still lacking in many areas. For example, there is no closed-loop feedback control for the illuminance level of the room and hence daylight harvesting cannot be employed without the use of external sensors. Some other concerns associated with smart lights such as Philips Hue and smart homes in general which integrate with the IoT are the issues of security and privacy [6, 7]. The Philips Hue system along with some other smart home appliances were studied to show some security loopholes [6] and were even successfully hacked into [7]. This raises a huge concern as hackers are then able to monitor the status of these smart homes removing all privacy and would even be able to control the lights causing a blackout to the entire house, leaving the owner of the house worse off than using the traditional lighting systems with wall switches.

As the research on smart homes advances, various smart home models and architectures have been proposed. One such research constructed their own home server to automate various home equipment [8]. Some others proposed modifying residential gateways in smart homes for home energy management system [9] or to connect the smart home to the cloud [10, 11]. There are also various connectivity options for smart homes, and the wireless networks are usually preferred over wired solutions as the wiring has to be planned for during the design and construction of the building and therefore the smart home system would not be able to be implemented in older buildings unless extensive renovation works are carried out. Common wireless technologies used are Bluetooth, Wi-Fi and the most

\*Corresponding author.

Email addresses: samuel.tang@monash.edu (Samuel Tang), vineetha@monash.edu (Vineetha Kalavally), ng.kok.yew@monash.edu (Ng Kok Yew), jussi.parkkinen@uef.fi (Jussi Parkkinen)

popular of all is ZigBee due to its low cost, power consumption and complexity [12]. Therefore, it has been the focus in many smart home applications such as the energy management system [13], the intelligent lighting control system [14, 15], and home automation [16].

When it comes to lighting control, in an effort to reduce power wastage on artificial lighting when the room is not occupied, past research has incorporated occupancy sensors into lighting systems to turn off the lights automatically [17]. Some existing work done on controlling room illuminance made use of manual inputs from users to adjust the illumination intensity [18], or by using a sensor network [19, 20]. So far, all research focusing on daylight harvesting and intelligent lighting control utilize sensor networks such as in [21, 22, 23, 24] and therefore would be costly to implement.

With minimal research done on smart home architecture for lighting, and commercial products having inadequacies, this paper proposes a smart home lighting system with enhanced security features together with a very economical solution for daylight harvesting using the user's personal smartphone.

In section 2, the smart home lighting architecture is proposed; in section 3, the security features implemented are discussed; in section 4, a novel closed-loop feedback control of room lighting using a smart phone is demonstrated; in section 5, the performance of the system is evaluated and the results of a daylight harvesting experiment is discussed; in section 6, a cost analysis and comparison of the system with similar commercial products is performed; and section 7 concludes the paper.

## 2. Smart Home Lighting Architecture

The three main components in the intelligent lighting system are the smartphone application, the luminaire and the main controller. The smartphone application provides the user interface for the user to control the lighting system and also to obtain the lux readings from the light sensor on the smartphone to perform closed-loop lighting feedback. The luminaire has red, green and blue LED channels controllable via pulse-width modulation (PWM) using an on-board Arduino Uno. A Raspberry Pi is used as the main controller which acts as the interface between the mobile application and the luminaires. It also acts as the home server for the user to connect to the system via the Internet while away from home and it could also be the platform in which other systems in smart homes can be integrated to in the future. These three main components interact with each other as shown in Figure 1. The luminaire and main controller of the intelligent lighting system is shown in Figure 2. The figure shows an eight-channel luminaire but only 3 channels were used in the setup and testing.

Since the human eye has photoreceptors at the red, green and blue frequencies, a three-channel luminaire with a dedicated channel for each of those 3 frequencies is sufficient to produce a significant number of colors differentiable by the human eye. Each channel is powered by a driver with variable current controlled by PWM having a range from 0–255. The Arduino Uno microcontroller controls the drivers and communicates with the

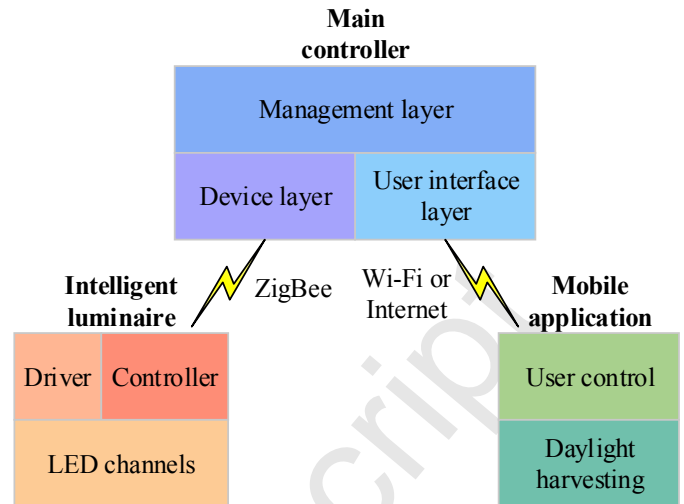


Figure 1: Architecture of the proposed smart home lighting control system

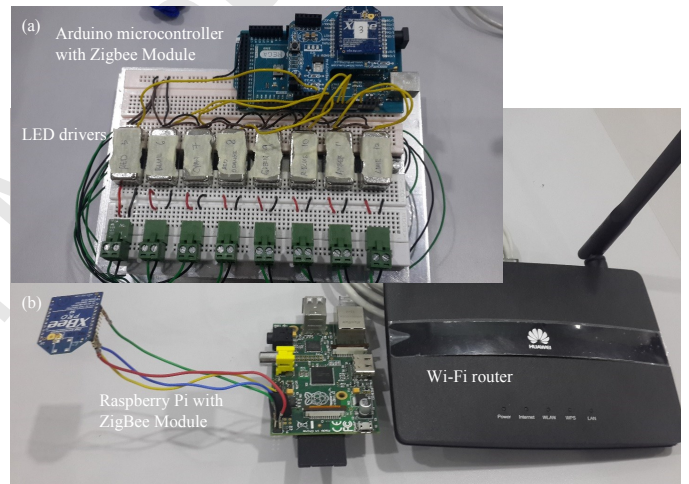


Figure 2: (a) Luminaire controlled by the Arduino microcontroller. (b) Main controller connected to the Wi-Fi router

main controller via an XBee shield which is attached to the Arduino board. Each of the XBee modules are programmed to have a different address for luminaire differentiation and the Application Programming Interface (API) mode 2 was used to ensure a reliable communication between the XBee modules. The data transmitted between the main controller and the luminaire consists of 4 bytes containing all the required information to control the luminaire and is as summarized in Table 1.

The read/write flag is used to determine if the main controller is requesting the current settings of the luminaire or is instructing the luminaire to update to the new data in the following bytes. The off/on flag is used to toggle the luminaire off/on and it is also used in the reply from the luminaire to the main controller to report on its current state. The change color flag is set to 1 if the main controller wants the luminaire to update its red, green and blue (RGB) values to the data in the

Table 1: Distribution of bits of the payload of the XBee Packet

Description	Size
Flags	
• Read/Write	1 bit
• Off/On	1 bit
• Change color	1 bit
• Unused	5 bits
Red value	1 byte
Green value	1 byte
Blue value	1 byte

packet, else it is set to 0. The red, green and blue values are the new PWM values that the luminaire should update to. Algorithm 1 shows the pseudocode of the microcontroller in communicating with the main controller and controlling the luminaire.

---

**Algorithm 1** Luminaire Arduino microcontroller pseudocode

---

```

loop
  while (Serial.available() > 0) do
    Serial.read()
  end while
  Unescape characters, check packet length and checksum
  if Valid packet then
    Extract flags
    if Write then
      if Change color then
        Extract new color values from packet
        Update PWM values
      end if
      Turn on or off the luminaire based on flag
    end if
    Create reply packet
    Reply
  end if
end loop

```

---

The Raspberry Pi model B was chosen over other devices as the main controller due to its relatively lower price compared to its competitors, and its ability to run on Linux. It also contains 26 dedicated general-purpose input-output (GPIO) pins, of which some are being used to connect to an XBee module communications with the luminaire and the remaining unused pins can be used to integrate with other smart home systems in the future if needed.

The software architecture of the main controller for this smart home intelligent lighting control is proposed to have the structure with the arrows depicting the direction of communication as shown in Figure 3. The Device Layer handles all the communication and protocols between the luminaires and the main controller, which includes packaging and unpackaging the XBee packets, checking for errors and timeouts. The Management Layer handles all the protocol conversions, extraction of

commands from JavaScript Object Notation (JSON) structures and passing them to the device layer and vice versa. User account management is also handled in this layer to enable or to restrict access of users to the system or certain luminaires. This layer handles automation, for example time-based functions such as turning on a group of lights at a certain time. The User Interface Layer serves as the interface between the main controller and the user. The Transmission Control Protocol (TCP) server is set up to receive connections from the smartphone and the Simple Service Discovery Protocol (SSDP) server is used to provide the smartphones with the IP address of the server. The SSDP is the basis of the discovery protocol of Universal Plug and Play (UPnP) services [25] and is used here so that the main controller would be able to have a dynamic IP address as allocated by the network to avoid conflicting Internet Protocol (IP) address complications. The implementation of the protocol follows the specification provided in the UPnP website and forum report [25]. The home server used to enable users to connect to the system even when they are not on the same network is also established in the User Interface Layer. Since the IP address of houses might change according to the Internet Service Providers, the NoIP service is used to handle the dynamic Domain Name System (DNS) so that the user is able to monitor and control the lighting system even when away from home. The router of the home system has to be configured to forward the required ports. All codes for the Raspberry Pi main controller are written in Python and all network sockets are run in new threads to ensure that the sockets would not cause the system to be unresponsive to new inputs. The flow of commands and data in the main controller when a user changes a setting of brightness, color or toggle of state is shown in Figure 4.

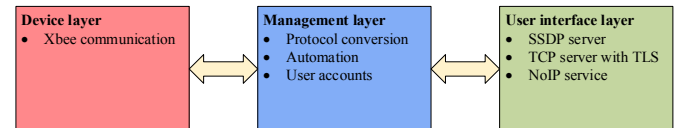


Figure 3: Communications between layers in the software architecture of the proposed system

Upon opening the application, the login screen is loaded and the IP address of the main controller is searched for in the background. Once logged in, the list of the luminaires is displayed with the options to change the brightness, color from a preset list or color wheel, and toggle on/off state. The closed-loop control which is run as a service in the background so that the user can use other applications on the smartphone without disrupting the feedback control is also started here. The user interface of the smartphone application is shown in Figure 5. The code to implement the color picker was adapted from a publicly shared code on GitHub [26].

### 3. Security Features

Security features are very important in the IoT to enhance the safety of the system and privacy of the user. The Public Key In-

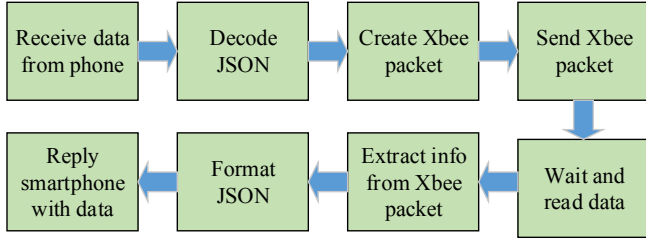


Figure 4: Flow of commands and data in the main controller when the luminaire setting is changed

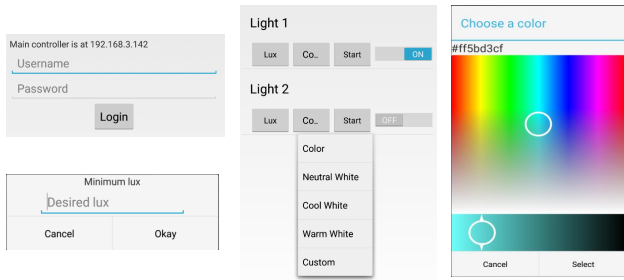


Figure 5: The user interface of the smartphone application: login screen (top left), pop-up to enter desired lux (bottom left), list of luminaires (middle), pop-up to choose desired color (right)

frastructure (PKI) is used to secure the connection between the smartphone and the main controller as suggested in the IEEE Standards Association [27]. The main controller runs the TCP server secured with Transport Layer Security (TLS) and for users to access the system, a login is required with a username and password, providing similar security features to Hypertext Transfer Protocol Secure (HTTPS) services thwarting attacks such as replays, man-in-the-middle and others. An X.509 self-signed certificate and public-private key pair were generated and stored in the main controller. A BouncyCastle (BKS) file is then generated and loaded into the custom KeyStore in the Android application. A custom HostNameVerifier is also created in the application to verify the identity of the server. With TLS implemented, all data and commands between the users and the main controller are secure and authentic even if the home network has been compromised. This would prevent the ‘perpetual blackout’ attack that Nitesh Dhanjani successfully implemented on the Philips Hue [28].

On the other hand, the XBee connection between the main controller and the luminaires are secured using the Advanced Encryption Standard (AES). In the event of new XBee devices being added into the system, the main controller can use the AT commands to connect to the new device and set the AES key, hence the system is flexible enough to handle the addition of new luminaires.

#### 4. Daylight Harvesting Using the Mobile Device

It is apparent that with the implementation of daylight harvesting, a substantial amount of energy can be saved. However, installing sensor networks in the home system would be very costly and since there are various sensors available on a typical smartphone which are ubiquitous nowadays, utilizing those sensors for the lighting system would prove to be very economical. One of the sensors used to perform daylight harvesting is the light sensor to detect ambient light for automatic screen brightness adjustment. On Android devices, this sensor returns a lux value and this is used to feedback to the lighting system to either dim or brighten the lights to consistently maintain the desired illuminance level.

The feedback control block diagram used to control the light intensity to balance daylight harvesting and user comfort, is shown in Figure 6 with the description of the variables in Table 2.

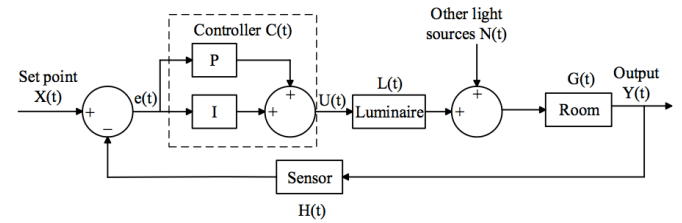


Figure 6: Feedback control block diagram of lux

Table 2: Description of variables in Figure 6

Variable	Description
$X(t)$	reference lux set by user
$e(t)$	error signal signifying the difference between the reference lux and actual sensed lux values
$C(t)$	PI controller
$U(t)$	control input for the Luminaire system, defined as $U(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$ , where $K_p$ and $K_i$ are the proportional and integral gains of the controller respectively, both tuned using the Ziegler-Nichols method
$L(t)$	transfer function of the Luminaire system
$N(t)$	other external unmodelled light sources in the room, e.g. sunlight through the window
$G(t)$	transfer function of the object surfaces in the room
$H(t)$	transfer function of the light sensor in the smartphone
$Y(t)$	illuminance level of the room as detected by the light sensor and perceived by the human eye

The error signal,  $e(t)$ , and the room illuminance level,  $Y(t)$ , can be expressed as,

$$e(t) = X(t) - H(t)Y(t), \quad (1)$$

$$Y(t) = (e(t)C(t)L(t) + N(t))G(t). \quad (2)$$



Combining equations (1) and (2) yield the output or the illuminance level of the room that the light sensor observes:

$$Y(t) = \frac{C(t)L(t)G(t)}{1+H(t)C(t)L(t)G(t)}X(t) + \frac{G(t)}{1+H(t)C(t)L(t)G(t)}N(t) \quad (3)$$

The proportional-integral (PI) controller is chosen so that the system is able to achieve zero steady-state error. The differentiator is not used as it would amplify existing noise in the system. There are some methods that have been proposed to smoothen the smartphone sensor readings [29], but these methods are not employed as they introduce a delay in the sensor reading which would cause the feedback control to react slowly.

From equation (3), it can be seen that daylight harvesting can be performed with this system as both the desired illuminance level and the other light sources are passed through the feedback loop and hence the output from the luminaire can be adjusted to meet the user's needs. When there is daylight illuminating the room, the feedback loop would cause the luminaire to dim, achieving a comfortable illuminance level for the user and reducing electrical power consumption at the same time. The only situation where this system would fail to meet the user's specification is when the user has selected a very high desired lux and the maximum output from the luminaire combined with daylight is not able to meet this demand; or when the user's desired illuminance level is lower than the incoming daylight, of which in this case the luminaire will be turned off entirely and the room will be lit only by daylight.

Algorithm 2 shows the pseudocode of the closed-loop feedback controller implemented on the Android smartphone using the ambient light sensor. The calculations are performed in the Hue Saturation Value (HSV) color space since the value component in the HSV model represents the brightness of the color. It is then converted into RGB values to be sent to the luminaires. The hue and saturation values of the luminaire are maintained for the selected color of the user so that daylight harvesting can be implemented while maintaining the color of the light being emitted from the luminaire.

Since the light sensor on the smartphone is capable of reading values faster than the time it takes to do all the calculations and communications within the devices, a `feedbackDone` flag is used in the algorithm to ensure each loop of the feedback is completed before the next step is performed. The light sensor is set to the slowest speed on Android using the setting `SensorManager.SENSOR_DELAY_NORMAL` to reduce usage of computing resources and power consumption. At this setting, a lux reading is returned on average every 179 ms. On the other hand, each feedback loop would perform the computations and two-way communications until the smartphone receives an acknowledgment message which takes approximately 423 ms to complete with a standard deviation of 119 ms. This relatively large standard deviation is due to timeouts in the communication links caused by loss or corrupted packets.

However, since the smartphone is being used as the sensor in the system, it might seem to be depriving the users from using their smartphones. Therefore, an algorithm using the accelerometer in the smartphone was written to detect the whenever the smartphone is being picked up and pauses the daylight

#### Algorithm 2 Closed-loop feedback control pseudocode

```

Get luminaire settings
onLuxReadingChanged()
if feedbackDone = true then
    feedbackDone ← false
    luxNow ← lux sensor reading
    timeChange ← system current - last loop time
    luxErr ← referenceLux - luxNow
    luxErrSum ← luxErrSum + luxErr × timeChange
    luxOutput ← (Kp × luxErr) + (Ki × luxErrSum)
    luxGain ← exp(luxOutput / selectedLux)
    newHsvValue ← currentHsvValue × luxGain
    if ((newHsvValue > 1)) then
        newHsvValue ← 1
    end if
    if (newHsvValue = 0 and luxGain > 1) then
        newHsvValue ← 0.004
    end if
    newHue, newSaturation ← currentHue, currentSaturation
    Convert HSV to RGB
    Send new values to main controller
    Wait and receive reply
    feedbackDone ← true
end if

```

► end of sensor callback

harvesting control loop. This allows the users to use their smartphones and not interfere with the system and after usage, the daylight harvesting algorithm can then be resumed. Since the smartphone can be placed at any angle in 3-dimensional space, the total acceleration,  $a_{total}$ , the smartphone is experiencing is calculated from the x, y, and z axis components.

$$a_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (4)$$

where  $a_x$ ,  $a_y$ , and  $a_z$  are the accelerations in the x, y, and z axes as shown in Figure 7.

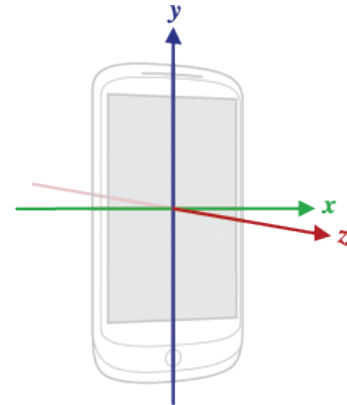


Figure 7: Coordinate system relative to the smartphone that is used by the sensor API

But the `Sensor.TYPE_ACCELEROMETER` in Android always has the force of gravity influencing the measured acceleration. Therefore, a high pass filter was implemented to remove the effects of gravity,  $a_g$ , to obtain only the quick motion of picking up a phone,  $a_{pick}$ . Once  $a_{pick}$  passes  $1m^{-2}$ , it is considered that the user picked up the phone and the daylight harvesting algorithm automatically pauses.

$$a_g(nT) = \alpha a_g[n(T-1)] + (1-\alpha)a_{total}(nT) \quad (5)$$

$$a_{pick}(nT) = a_{total}(nT) - a_g(nT) \quad (6)$$

Where  $n$  is the sample number,  $T$  is the sampling period, and  $\alpha$  is the filter constant.

Additionally, other Android APIs can be used to detect usage of the smartphone or when an event will require the daylight harvesting to pause such as receiving a phone call. A sampling period setting can also be included to allow the user to adjust the sampling period when the system has already converged to the desired illuminance level. Lowering the sampling period when the room lighting condition is expected to only change slightly or gradually will reduce battery power consumption.

## 5. Experimental Results and Discussions

Any Android smartphone nowadays is capable of running this algorithm as the minimum Android version required is Android 1.5 and contain a working ambient light sensor which is a common sensor available in all typical smartphones. The Samsung Galaxy S4 was used for testing as it was one of the latest smartphone models in the market at that time. The light sensor readings from the smartphone were compared against the Konica Minolta CL-500A illuminance spectrophotometer under various light sources such as fluorescent U30, TL84 as well as the RGB luminaire being set to warm, neutral and cool white. The light source illuminance was varied to produce between 100 and 1000 lux and the smartphone light sensor average percentage error was computed using equation 7 where  $E$  is illuminance. The results are tabulated in Table 3.

$$Error = \frac{E_{sensor} - E_{spectrophotometer}}{E_{spectrophotometer}} \times 100\% \quad (7)$$

Table 3: Accuracy of smartphone light sensor with respect to Konica Minolta CL-500A illuminance spectrophotometer

Light source	Average percentage error	Standard deviation
U30	5.543	1.624
TL84	2.892	1.072
Warm white	2.348	4.281
Neutral white	-0.014	4.411
Cool white	-2.529	3.225

The amount of sensor noise in the light sensor was also tested under warm, neutral and cool white and the results are tabulated in Table 4.

Table 4: Light sensor noise for different lighting temperature and illuminance level

Light source and illuminance level	Standard deviation of percentage error
Warm white (1144 lux)	1.382
Warm white (585 lux)	0.892
Neutral white (1005 lux)	1.154
Neutral white (517 lux)	1.039
Cool white (1012 lux)	1.766
Cool white (504 lux)	1.966

The relatively high accuracy and low sensor noise in the sensor readings show that the smartphone light sensor is suitable to be used in smart homes for daylight harvesting.

The closed-loop feedback algorithm was then tested using a step response to determine how well it responds to disturbances. A large step disturbance of 1000 lux to 400 lux and 400 lux to 1000 lux and a small step disturbance of 500 lux to 600 lux and 600 lux to 500 lux were used. It can be seen from Figure 8 that the system responded well to the disturbances and converged to the desired illuminance level within 3 oscillations or 1.5 seconds. In actual applications, disturbances are smaller and more gradual, usually represented by phenomena such as clouds moving across the sky or the setting of the sun. Therefore this rate of convergence is sufficient to ensure a constant illuminance level in the room.

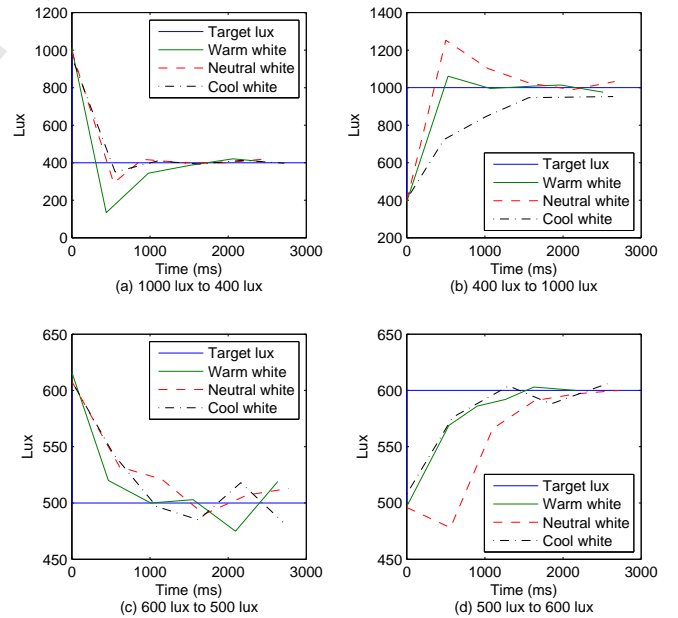


Figure 8: Step responses of the closed-loop controller

Figure 9 shows in percentage power the power consumption of the system compared to having the luminaire being turned on to produce 500 lux the whole day. 500 lux was chosen as the reference level as this meets the recommended level for most indoor work places as stated in the joint ISO/CIE standard [30]



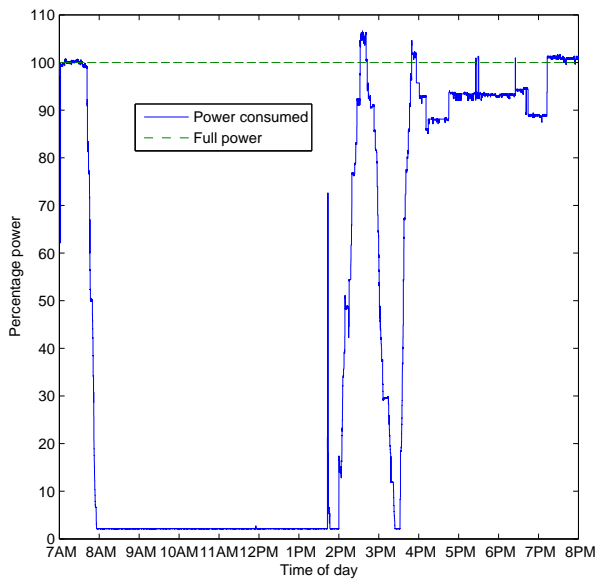


Figure 9: Power consumption of lighting system implementing daylight harvesting

which is generally brighter than home illuminance levels. This is a typical day in Malaysia where the sun starts to rise at around 7:30am, and since the room was facing East, it received a lot of sunlight in the morning. And very commonly from around 2pm onwards, there is heavy rain intermittently and the sun sets at around 7:30pm. With this system in place, a power savings of 54.7% was achieved. It is to note that this result was achieved for mostly a sunny day condition (except in the afternoon) and that the mock living room has big windows and the smartphone was placed beneath the luminaires and beside the window as shown in Figure 10.

The light sensor is quite directional and therefore has to be placed at an angle and position that is able to detect both the sunlight coming in from the windows as well as the light from the luminaires. Hence, one inconvenience of using the smartphone instead of a dedicated physical sensor network is that the user is required to place the smartphone correctly to optimize daylight harvesting. Also, any hand-held usage of the smartphone would pause the daylight harvesting algorithm for a short while but it should not be a huge issue as daylight intensity changes gradually and therefore a short usage of the mobile phone would not affect the system much. And just like any other physical sensor networks placed at the working level or table height, people and other objects moving within the room might cast a shadow on the light sensor, causing undesired luminance fluctuations. However, this effect can be decreased by an adjustable parameter damping the response to the pulse changes in illumination.



Figure 10: Photo of the mock living room

## 6. Cost Analysis

The cost of the intelligent lighting system components are presented in Table 5. The base cost to setup the main controller (Raspberry Pi, wireless router, Ethernet cable, XBee module) is USD 59.44 and each luminaire costs USD 49.17. For comparison, imagine setting up a room comprising of 3 luminaires. The cost of using this system and the cost of additional luminaires is compared with similar wireless RGB luminaire systems in the market such as the Philips Hue, LIFX and Osram LIGHTIFY is compared in Figure 11. It is to take note that the cost of the wireless router has not been included in the calculation because most homes would already have a wireless router which could be used instead, and the Hue and LIGHTIFY costs do not include it as well.

Though it seems that the proposed system is about the same price as compared to commercial products, it is because the commercial luminaires are rated at 10 watts while the luminaire used here goes up to 40 watts which is a significant difference. The use of the Raspberry Pi in this system also opens up the possibility of implementing new features or integrating this system to any other IoT system in the future which could be very advantageous as compared to the current market products. The Raspberry Pi model 1 B could also be substituted with another similar computer such as the Orange Pi at just a quarter the cost and the Arduino Uno could also be reduced to a cheaper model since only 3 pins are required to control the RGB chan-

Table 5: Cost of the intelligent lighting system

Item	Cost (USD)	No.	Total Cost (USD)
Arduino Uno	5.00	1	5.00
XBee Module	1.00	2	2.00
XBee shield	4.00	1	4.00
PCB cost	2.37	1	2.37
RCD-24-0.70 LED driver	7.60	3	22.80
Cree XB-D LEDs	1.00	14	14.00
HuaWei WS319 wireless router	17.00	1	17.00
Raspberry Pi 1 B	39.95	1	39.95
RJ45 Ethernet cable	1.49	1	1.49
<b>Total</b>			<b>108.61</b>

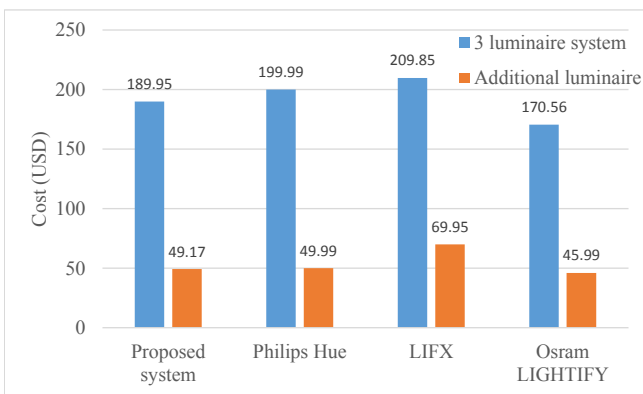


Figure 11: Comparison of cost with equivalent systems in the market without daylight harvesting capability

nels. Although there are other costs involved in the commercial products, the cost of this system can still be reduced through optimization of the luminaire circuitry and mass production. Most importantly, the commercial products do not have daylight harvesting capability which would result in higher savings in the long run.

To further analyze the cost-saving potential of this system, the mock living room having a floor area of 5.8 meters by 3.4 meters is analyzed. Assuming the amount of daylight coming in on average is similar to Figure 9 as some days would be more sunny and some days more cloudy, and the living room is being used the whole day at an electricity tariff of MYR 0.571 per kWh or USD 0.14 per kWh with 6 luminaires installed. The setup cost and cost-saving potential is summarized in Table 6. If the system was more optimized and cheaper components are used instead, the time taken for the cumulative cost savings to exceed the setup cost would be much lower.

Even if this intelligent lighting system is not implemented, the daylight harvesting algorithm on the smartphone can still be adapted to be used in other wirelessly controlled lighting systems such as the ones available in the market. This would still enable the system to perform daylight harvesting without the installation of physical sensors and hence prove to be very

Table 6: Daylight harvesting savings potential

Setup cost	USD 337.46
Potential savings per month	USD 5.38
Time taken for cumulative cost savings to recover setup cost	62.8 months

convenient and economical.

## 7. Conclusion

This paper has presented the design and prototyping of a smart home lighting system with effective and efficient daylight harvesting capability. The system comprises an Arduino controlled luminaire with RGB channels, a Raspberry Pi main controller and a mobile application on the user's smartphone. Users are able to control individual or multiple luminaires conveniently and securely using their mobile devices at home through the local Wi-Fi network or the Internet. The light sensor on the Samsung Galaxy S4 has been tested to show that it produces high accuracy and low noise lux readings under white light. Therefore, a novel closed-loop feedback system using the light sensor on the smartphone was created enabling users to perform daylight harvesting while maintaining their desired lighting color. A daylight harvesting experiment was run in the mock living room in Monash University Malaysia and it achieved power savings of up to 54.7% on a typical Malaysian day. A cost analysis shows that this system has more potential and is slightly cheaper than commercial products, and with its daylight harvesting capabilities, it can recover its setup cost in 62.8 months. This concept design proves to be a very economical and convenient alternative to installing additional physical sensors.

## Acknowledgements

Funding: This research was supported by ItraMAS Sdn Bhd, Malaysia.

## References

- [1] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, X. Lin, Smart community: an internet of things application., IEEE Communications Magazine 49 (11) (2011) 68–75.
- [2] T. Baumgartner, F. Wunderlich, A. Jaunich, T. Sato, G. Bundy, N. Griebmann, J. Kowalski, S. Burghardt, J. Hanebrink, Lighting the way: Perspectives on the global lighting market.
- [3] I. E. Agency, Energy efficiency market report 2015.
- [4] J. D. Jennings, F. M. Rubinstein, D. DiBartolomeo, S. L. Blanc, Comparison of control options in private offices in an advanced lighting controls testbed, Journal of the Illuminating Engineering Society 29 (2) (2000) 39–60.
- [5] Y.-J. Wen, J. Granderson, A. M. Agogino, Towards embedded wireless-networked intelligent daylighting systems for commercial buildings, in: Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on, Vol. 1, IEEE, 2006, pp. 6–pp.
- [6] B. Ur, J. Jung, S. Schechter, The current state of access control for smart devices in homes, in: Workshop on Home Usable Privacy and Security (HUPS), 2013.

- [7] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, R. Boreli, An experimental study of security and privacy risks with emerging household appliances, in: Communications and Network Security (CNS), 2014 IEEE Conference on, IEEE, 2014, pp. 79–84.
- [8] I. Han, H.-S. Park, Y.-K. Jeong, K.-R. Park, An integrated home server for communication, broadcast reception, and home automation, in: Consumer Electronics, 2006. ICCE'06. 2006 Digest of Technical Papers. International Conference on, IEEE, 2006, pp. 253–254.
- [9] D.-M. Han, J.-H. Lim, Design and implementation of smart home energy management systems based on zigbee, Consumer Electronics, IEEE Transactions on 56 (3) (2010) 1417–1425.
- [10] Z. Wei, J. Li, Y. Yang, D. Jia, A residential gateway architecture based on cloud computing, in: Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on, IEEE, 2010, pp. 245–248.
- [11] D. Valtchev, I. Frankov, Service gateway architecture for a smart home, Communications Magazine, IEEE 40 (4) (2002) 126–132.
- [12] I. Chew, V. Kalavally, C. P. Tan, J. Parkkinen, A spectrally tunable smart led lighting system with closed-loop control, IEEE Sensors Journal 16 (11) (2016) 4452–4459.
- [13] D.-M. Han, J.-H. Lim, Smart home energy management system using ieee 802.15. 4 and zigbee, Consumer Electronics, IEEE Transactions on 56 (3) (2010) 1403–1410.
- [14] T. Wang, B. Zheng, Z.-L. Liang, The design and implementation of wireless intelligent light control system base on zigbee light link, in: Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2013 10th International Computer Conference on, IEEE, 2013, pp. 122–125.
- [15] J. Yao, Z. Zhang, P. Dong, K. Holtman, A zigbee controlled lighting system with improved resistance to wireless interference, in: Wireless and Mobile, 2014 IEEE Asia Pacific Conference on, IEEE, 2014, pp. 219–225.
- [16] K. Gill, S.-H. Yang, F. Yao, X. Lu, A zigbee-based home automation system, Consumer Electronics, IEEE Transactions on 55 (2) (2009) 422–430.
- [17] A. Fernández-Montes, L. Gonzalez-Abril, J. A. Ortega, F. V. Morente, A study on saving energy in artificial lighting by making smart use of wireless sensor networks and actuators, IEEE network 23 (6) (2009) 16–20.
- [18] J. Byun, I. Hong, B. Lee, S. Park, Intelligent household led lighting system considering energy efficiency and user satisfaction, IEEE Transactions on Consumer Electronics 59 (1) (2013) 70–76. doi:10.1109/TCE.2013.6490243.
- [19] Y. K. Tan, T. P. Huynh, Z. Wang, Smart personal sensor network control for energy saving in dc grid powered led lighting system, Smart Grid, IEEE Transactions on 4 (2) (2013) 669–676.
- [20] M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, Y.-C. Tseng, A wsn-based intelligent light control system considering user activities and profiles, Sensors Journal, IEEE 8 (10) (2008) 1710–1721.
- [21] A. Pandharipande, D. Caicedo, Adaptive illumination rendering in led lighting systems, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 43 (5) (2013) 1052–1062.
- [22] M. T. Koroglu, K. M. Passino, Illumination balancing algorithm for smart lights, Control Systems Technology, IEEE Transactions on 22 (2) (2014) 557–567.
- [23] Y. Uhm, I. Hong, G. Kim, B. Lee, S. Park, Design and implementation of power-aware led light enabler with location-aware adaptive middleware and context-aware user pattern, Consumer Electronics, IEEE Transactions on 56 (1) (2010) 231–239.
- [24] A. Pandharipande, D. Caicedo, X. Wang, Sensor-driven wireless lighting control: System solutions and services for intelligent buildings, Sensors Journal, IEEE 14 (12) (2014) 4207–4215.
- [25] U. Forum, Upnp device architecture 2.0.
- [26] jbruchanov, Androidcolorpicker (2015).  
URL <https://github.com/jbruchanov/AndroidColorPicker/tree/master>
- [27] Ieee standard for ubiquitous green community control network: Security, IEEE Std 1888.3-2013 (2013) 1–30doi:10.1109/IEEESTD.2013.6675755.
- [28] N. Dhanjani, Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts, "O'Reilly Media, Inc.", 2015.
- [29] R. Ozcan, F. Orhan, M. F. Demirci, O. Abul, An adaptive smoothing method for sensor noise in augmented reality applications on smart-phones, in: Mobile Wireless Middleware, Operating Systems, and Applications, Springer, 2011, pp. 209–218.
- [30] S. CIE, 008/e: 2001: Joint iso/cie standard: Lighting of work places–part 1: Indoor [incl. technical corrigendum iso 8995: 2002/cor. 1: 2005 (e)], Vienna, Austria: Commission Internationale de LEclairage.